Tensorflow Classification Server

Functionality & Features:

Server has 6 API endpoints in the server.

1. http://localhost:5000/get_train_status [GET]:
    a. We track all the files on which our latest model is trained. This API will return the number of files in each class on which model is trained and number of files which are added after model training, example response:
    {'daisy': {'tracked': 633},
    'dandelion': {'tracked': 898},
    'roses': {'tracked': 641},
    'sunflowers': {'tracked': 699, 'untracked': 40},
    'tulips': {'tracked': 799}})

    Above example shows that 40 images in roses are added after the latest model training.

2. http://localhost:5000/train?epochs=10 [GET]:
    a. Used to start the training. We need to pass number of epoch i.e. 10 in above example. The program will treat each folder in assets as a separate class e.g.

    ---------assets
    ---------- cats
    ---------- dogs
    ---------- horse

    b. In above scenario, classifier will be trained on these 3 classes cats, dogs, horse. We can train as many models as we want. Each model will be stored as current timestamp in its name. e.g. 2020-07-12_04:38:49_model
    c. We can log into server to see the live performance of training e.g.

3. http://localhost:5000/list_models[GET]:
    a. This API will list all the trained models with their performance metrics e.g. accuracy, loss, etc. Example response:

```
{'2020-07-12_04:39:36_model.h5': {
        'accuracy': 0.7510302066802979,
        'is_default': True,
         'loss': 0.94694721698761,
         'val_accuracy': 0.8380681872367859,
         'val_loss': 0.7858497500419617},
 '2020-07-12_06:21:23_model.h5': {
        'accuracy': 0.7486263513565063,
        'is_default': False,
        'loss': 0.9259734749794006,
        'val_accuracy': 0.8536931872367859,}
        'val_loss': 0.7246572971343994}
}
```

4. http://localhost:5000/select_model?model_name=2020-07-12_06:21:23_model [GET]:
    a. By default, model with highest accuracy is selected to do the predictions. But we can select our default model for predictions. First we need to call list_models API and then we need to pass the model name select_model API. The selected model will do all the future predictions. The list_models API response has a key 'is_default', which shows the default selected model.

5. http://localhost:5000/classify [POST]:
    a. To classify image.
        i. Example call:
        curl -X POST -F image=@test.jpg 'http://localhost:5000/classify'
        ii. Example response:
        {'y_pred': 'sunflowers', 'probs': "{'daisy': 0.087750435, 'dandelion': 0.3376642, 'roses': 0.1948547, 'sunflowers': 0.8982313, 'tulips': 0.030957516}"}

    API returns the predicted class in y_pred key and probabilities of each class.

6. http://localhost:5000/add_train_img [POST]:
    a. Used to add train image into our train dataset.
    b. The name of the image file should be the name of class to which this image belongs. E.g.
    curl -X POST -F image=@assets/dogs.jpg 'http://localhost:5000/add_train_img'
    This above call will add new image to dogs class folder in assets.
    c. We can call get_train_status API to check number of newly added images in our dataset.